

CLAIM LISTING

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method for controlling code in a multi-developer software development environment, the method comprising:

~~identifying a plurality of software components and preventing access and modification to the plurality of non-modifiable software components;~~

receiving a request from a requesting user to ~~modify one of the~~ check out a software components component;

tagging the software component as non-modifiable by the requesting user and preventing modification to the software component by the requesting user;

determining whether the software component has been checked out by another user; and

untagging the software component, providing the requesting user with a modifiable copy of the one of the software component, and receiving a description of changes intended to be made by the requesting user if the software component has not been checked out by another user.

2. (Currently Amended) The method as recited in claim 1, further comprising:

presenting the requesting user with an indication that the software component has been checked out by another user and is

not available for modification if the ~~object~~ software component has been checked out by another user.

3. (Original) The method as recited in claim 1, further comprising:

creating a backup copy of the software component prior to providing the user with a modifiable copy of the software component.

4. (Original) The method as recited in claim 1, further comprising:

updating a file indicating that the software component is checked out, the identity of the user that checked out the software component, the date the software component was checked out, and the time the software component was checked out.

5. (Cancelled)

6. (Original) The method as recited in claim 5, further comprising:

updating a file indicating the changes intended to be made to the software component.

7. (Original) The method as recited in claim 1, wherein the software component is an object.

8. (Original) The method as recited in claim 4, wherein the file is one of a table and a database.

9. (Original) The method as recited in claim 6, wherein the file is one of a table and a database.

10. (Currently Amended) The method as recited in claim 2, further comprising:

determining whether the requesting user wishes to send a message to the user who currently has the ~~object~~ software component checked out;

prompting the requesting user to indicate the message to be sent to the user who currently has the software component checked out; and

sending the message to the user who currently has the software component checked out.

11. (Original) The method as recited in claim 10, further comprising:

determining the method the requesting user wished the message to be sent to the user who currently has the software component checked out; and

sending the message by the method specified by the requesting user.

12. (Original) The method as recited in claim 1, further comprising:

determining that the requesting user requests to check the software component back in;

prompting the requesting user to enter a description of what has changed to the software component; and

saving an updated software component, indicating that the software component is checked in, and indicating that the software component is not modifiable.

13. (Original) The method as recited in claim 12, further comprising:

updating a file indicating that the software component is checked back in, the date and time checked in, the identity of the developer who checked the software component back in, and the nature of the changes made to the software component.

14. (Currently Amended) A computer program product in a computer readable media for use in a data processing system for controlling code in a multi-developer software development environment, the computer program product comprising:

~~first instructions for identifying a plurality of software components and preventing access and modification to the plurality of non-modifiable software components;~~

~~second first instructions for receiving a request from a requesting user to modify one of the check out a software components component;~~

second instructions for tagging the software component as non-modifiable by the requesting user and preventing modification to the software component by the requesting user;

third instructions for determining whether the software component has been checked out by another user; and

fourth instructions for untagging the software component, providing the requesting user with a modifiable copy of ~~the one~~ of the software component, and receiving a description of changes intended to be made by the requesting user if the software component has not been checked out by another user.

15. (Currently Amended) The computer program product as recited in claim 14, further comprising:

fifth instructions for presenting the requesting user with an indication that the software component has been checked out by another user and is not available for modification if the ~~object~~ software component has been checked out by another user.

16. (Original) The computer program product as recited in claim 14, further comprising:

fifth instructions for creating a backup copy of the software component prior to providing the user with a modifiable copy of the software component.

17. (Original) The computer program product as recited in claim 14, further comprising:

fifth instructions for updating a file indicating that the software component is checked out, the identity of the user that checked out the software component, the date the software component was checked out, and the time the software component was checked out.

18. (Cancelled)

19. (Original) The computer program product as recited in claim 18, further comprising:

sixth instructions for updating a file indicating the changes intended to be made to the software component.

20. (Original) The computer program product as recited in claim 14, wherein the software component is an object.

21. (Original) The computer program product as recited in claim 17, wherein the file is one of a table and a database.

22. (Original) The computer program product as recited in claim 19, wherein the file is one of a table and a database.

23. (Currently Amended) The computer program product as recited in claim 15, further comprising:

sixth instructions for determining whether the requesting user wishes to send a message to the user who currently has the object software component checked out;

seventh instructions for prompting the requesting user to indicate the message to be sent to the user who currently has the software component checked out; and

eighth instructions for sending the message to the user who currently has the software component checked out.

24. (Currently Amended) The computer program product as recited in claim 23, further comprising:

ninth instructions for determining the ~~computer program~~
~~product~~ method the requesting user ~~wished~~ wishes the message to
be sent to the user who currently has the software component
checked out; and

tenth instructions for sending the message by the ~~computer~~
~~program-product~~ method specified by the requesting user.

25. (Original) The computer program product as recited in
claim 14, further comprising:

fifth instructions for determining that the requesting user
requests to check the software component back in;

sixth instructions for prompting the requesting user to
enter a description of what has changed to the software
component; and

seventh instructions for saving an updated software
component, indicating that the software component is checked in,
and indicating that the software component is not modifiable.

26. (Original) The computer program product as recited in
claim 25, further comprising:

eighth instructions for updating a file indicating that the
software component is checked back in, the date and time checked
in, the identity of the developer who checked the software
component back in, and the nature of the changes made to the
software component.

27. (Currently Amended) A system for controlling code in a multi-developer software development environment, the system comprising:

~~first means for identifying a plurality of software components and preventing access and modification to the plurality of non-modifiable software components;~~

~~second first means for receiving a request from a requesting user to modify one of the check out a software components component;~~

second means for tagging the software component as non-modifiable by the requesting user and preventing modification to the software component by the requesting user;

third means for determining whether the software component has been checked out by another user; and

fourth means for untagging the software component, providing the requesting user with a modifiable copy of ~~the one~~ of the software component, and receiving a description of changes intended to be made by the requesting user if the software component has not been checked out by another user.

28. (Currently Amended) The system as recited in claim 27, further comprising:

fifth means for presenting the requesting user with an indication that the software component has been checked out by another user and is not available for modification if the ~~object~~ software component has been checked out by another user.

29. (Original) The system as recited in claim 27, further comprising:

fifth means for creating a backup copy of the software component prior to providing the user with a modifiable copy of the software component.

30. (Original) The system as recited in claim 27, further comprising:

fifth means for updating a file indicating that the software component is checked out, the identity of the user that checked out the software component, the date the software component was checked out, and the time the software component was checked out.

31. (Cancelled)

32. (Original) The system as recited in claim 31, further comprising:

sixth means for updating a file indicating the changes intended to be made to the software component.

33. (Original) The system as recited in claim 27, wherein the software component is an object.

34. (Original) The system as recited in claim 30, wherein the file is one of a table and a database.

35. (Original) The system as recited in claim 32, wherein the file is one of a table and a database.

36. (Currently Amended) The system as recited in claim 28, further comprising:

sixth means for determining whether the requesting user wishes to send a message to the user who currently has the ~~object~~ software component checked out;

seventh means for prompting the requesting user to indicate the message to be sent to the user who currently has the software component checked out; and

eighth means for sending the message to the user who currently has the software component checked out.

37. (Currently Amended) The system as recited in claim 36, further comprising:

ninth means for determining the ~~system method~~ with which the requesting user ~~wished~~ wishes the message to be sent to the user who currently has the software component checked out; and

tenth means for sending the message by the ~~system method~~ specified by the requesting user.

38. (Original) The system as recited in claim 27, further comprising:

fifth means for determining that the requesting user requests to check the software component back in;

sixth means for prompting the requesting user to enter a description of what has changed to the software component; and

seventh means for saving an updated software component, indicating that the software component is checked in, and indicating that the software component is not modifiable.

39. (Original) The system as recited in claim 38, further comprising:

eighth means for updating a file indicating that the software component is checked back in, the date and time checked in, the identity of the developer who checked the software component back in, and the nature of the changes made to the software component.